WHAT IS CLAIMED IS:

1. A computing system comprising:

at least part of a memory hierarchy; and

a processor that stores plural execution contexts in a pipeline thereof, the processor performing a context switch between a first one and a second one of the execution contexts by freezing the first execution context in the pipeline and resuming execution using previously frozen state corresponding to the second execution context, the context switching performed without draining the first execution context from the pipeline.

2. The computing system of claim 1, further comprising:

a second pipeline of the processor, the processor performing the context switch in both the pipeline and the second pipeline.

3. The computing system of claim 1,

wherein the first and second execution contexts correspond to respective threads of a single multi-threaded application.

4. The computing system of claim 1,

wherein the first and second execution contexts correspond to distinct programs executing on the processor.

5. The computing system of claim 1, further comprising:

context selectable storage distributed throughout the pipeline, the context selectable storage coupled into the pipeline to represent intermediate pipeline states for at least two concurrently executing execution contexts.

6. The computing system of claim 5,

wherein at least some of the context-selectable storage distributed throughout the pipeline employs multi-bit flip-flops, wherein respective bits of

each multi-bit flip-flop correspond to a selectable one of the execution contexts.

7. The computing system of claim 1, further comprising:
a context-selectable register file coupled to the pipeline to represent architectural states for at least two concurrently executing execution contexts.

8. The computing system of claim 7,
wherein the context switch is performed without saving and restoring the execution contexts to and from the register file.

9. The computing system of claim 7,
wherein the memory hierarchy includes cache defined on die with the processor.

10. The computing system of claim 7,
wherein the memory hierarchy includes memory coupled to the processor via at least one bus.

11. A method of operating a processor, the method comprising:
executing plural execution contexts in a pipeline of the processor; and
performing a context switch between a first one and a second one of the execution contexts by freezing the first execution context in the pipeline and resuming execution using previously frozen state corresponding to the second execution context, the context switching performed without draining the first execution context from the pipeline.

12. The method of claim 11, further comprising:
detecting an exception condition and initiating the context switch in response thereto.

13. The method of claim 11, further comprising:

performing the context switch without saving and restoring the execution contexts to and from a register file.

14. The method of claim 11, further comprising:

maintaining a context-selectable register file coupled to the pipeline to represent architectural states for at least two concurrently executing execution contexts.

15. A processor comprising:

at least one pipeline, including storage distributed throughout the pipeline for at least two concurrently executing execution contexts, the processor supporting a context switch between a first one and a second one of the execution contexts by freezing the first execution context in the pipeline and resuming execution using previously frozen state corresponding to the second execution context, the context switching performable without draining the first execution context from the pipeline; and

a context-selectable register file coupled to the pipeline to represent architectural states for at least the two concurrently executing execution contexts.

16. A method comprising:

simultaneously representing throughout a processor pipeline, state information corresponding to plural active execution contexts; and

switching between a first one and a second one of the active execution contexts by freezing the first execution context in the pipeline and resuming execution using previously frozen state corresponding to the second execution context, the switching performed without draining the first execution context from the pipeline.

17. An apparatus comprising:

a processor coupled to at least part of a memory hierarchy; and

means defined in the processor for storing plural execution contexts in a pipeline thereof, the processor performing a context switch between a

first one and a second one of the execution contexts by freezing the first execution context in the pipeline and resuming execution using previously frozen state corresponding to the second execution context, the context switching performed without draining the first execution context from the pipeline.

18. An method of making a processor integrated circuit product, the method comprising:

defining a pipelined processor; and

fabricating the pipelined processor as an integrated circuit with in-pipeline storage for plural execution contexts thereof, the in-pipeline storage allowing a context switch between a first one and a second one of the execution contexts by freezing the first execution context in the pipeline and resuming execution using previously frozen state corresponding to the second execution context, the context switching performable in the fabricated pipelined processor without draining the first execution context from the pipeline.

19. The method of claim 18,

fabricating the in-pipeline storage without multiplexers and using multi-bit storage logic that substantially maintains an integrated circuit footprint corresponding to a pipeline with single-bit storage logic.